CSOUND Granular Synthesis Unit Generator

Allan S C Lee
Queen's University of Belfast
Rm 6.8B Ashby Building, Belfast BT9 5AH Northern Ireland
Tel: +44 (0)1232 33 5473 Fax: +44 (0)1232 667023 Email: ASC.Lee@ee.qub.ac.uk

ABSTRACT: Two CSOUND Granular Synthesis Unit Generators have been developed to provide an easy way to compose music with granular synthesised sound. The first unit generator provides low level control of every parameter such as the amplitude, grain size, gap size, pitch shift and envelope. All these parameters can be changed at control rate. The second unit generator provides a high level control. Multiple voices and random offset of parameters are built into the generator. A maximum of 128 voices in each generator is implemented but this can be increased by changing a parameter in the include file. Extremely high grain density can be produced. Stereo or quad output can be produced by using two or four unit generator statements with different seeds for the random number generator.

Introduction

As described in the paper "Introduction to Granular Synthesis" by Curtis Roads, the theory of granular synthesis was initially suggested by Dennis Gabor in 1947. Since 1971, many different techniques have been used to synthesise sounds using grains by many authors such as Lippe, Roads and True. These techniques range from dedicated software to custom built digital signal processing hardware. Granular synthesis using CSOUND can be achieved by using conditional statements or a combination of existing unit generators but it is very complicated and a very large orchestra / score file is needed to generate a short output. Two unit generators have been developed to provide different levels of control of synthesis. They are regular CSOUND unit generators which means they can be fully integrated into normal CSOUND compositions. A CSOUND function table is used as source, an unlimited type of sounds ranging from simple FM timbre to sampled sound can be used as the source for the grains.

Implementation

Granular synthesis can be classified as a form of additive synthesis. A high number of grains are added together to produce output. A grain is a signal with an amplitude envelope and the envelope can be any shape. For computation efficiency, linear attack and decay are implemented. An optional function table for generating different envelope shapes is also implemented. The first unit generator, grains3 provides low level control of the synthesis. Every parameter such as the amplitude, grain size, gap size, pitch shift, attack and decay of the envelope can be changed at control rate. It produces a single stream of grains with gaps or overlap between successive grains.

\[
\text{grains3} \quad \text{xamp} \quad \text{kpskip} \quad \text{fss} \quad \text{kgskip} \quad \text{kgsize} \quad \text{katt} \quad \text{kdec} \\
\text{wave table size read is by gen01 in no. of bytes} \\
\text{kskip in seconds} \quad \text{kgsize in seconds} \\
(\text{kskip} + \text{kgsize}) * \text{est} < \text{table size} \\
\text{kgsize} \\
\text{xamp} \\
\text{katt in %} \quad \text{100 %} \\
0 \leq \text{katt} < 100 % \\
0 \leq \text{kdec} < 100 % \quad (\text{katt} + \text{kdec}) < 100 %
\]

230  I C M C  P R O C E E D I N G S  1 9 9 5
A single `grains3` statement will generate a stream of grains as shown below. Multiple `grains3` statements can be used in an orchestra file to produce output and different streams can be assigned to different channels for stereo or quad output. The `kpshift` parameter controls the amount of pitch shift relative to the original sound sample using a time scaling technique with linear interpolation between data points. All k parameters can be modified by a random value generated by the `randk` unit generator. The second unit generator, `grains4`, provides a high level control of the synthesis. Multiple voices and random offset of parameters are built into the generator. Extremely high grain density can be produced with a single `grains4` statement.

```
grain grain grain grain grain grain

| gap | overlap | gap | gap | gap | gap |

---

`grains4` command: `grains4 samp invoice iratio imode lthd lfn lpshift lgskip igskip_os ilenlength kgap as kgsz size igsize_os att tdec [lised] [lpitch1] [lpitch2] [lpitch3] [lpitch4] [fines]`

invoice defines the number of voices (streams) to be generated. iratio defines the speed of the sampling pointer moving along the function table. A value of 0.1 will stretch the original sample 10 times. lthd defines the thresholding value; samples in the function table below this value will be skipped. lpshift - pitch shift control. If lpshift is 0, pitch will be set randomly up and down an octave. If lpshift is 1, 2, 3 or 4, up to four different pitches can be set amount the number of voices defined in invoice. The optional parameters lgskip1, lgskip2, lgskip3 and lgskip4 are used to adjust the pitch shifts. The optional parameter lgsizem defines the shape of the envelope of the grains. The default value is 0 and linear attack and decay is used. A positive value will be interpreted as a function table number and the data stored in the function table is used to generate the attack curve of the envelope. The decay curve will be a mirrored image of the attack curve. lseed defines the seed to be used for the random number generator. For multi-channel output, use different `lseed` values in each `grains4` statement. (Due to lack of space in this paper, full documentation will be available by anonymous ftp)

Conclusions and future work

The two unit generators have been compiled and tested on Sun, NeXt and SGI workstations. Unfortunately, the process is quite computationally intensive, generating a high number of voices is quite time consuming but the results are very rewarding. Very rich harmonic and spatial depth can be achieved by using 20 or more per channel. For fast turn-on, set invoice to 20 or less to experiment with different sound sources and parameters; set invoice to 64 or more for a production run.

Different shapes of grain envelope are under test at the moment to study how envelope shapes will affect the output in terms of harmonics and other perceptual factors.

Acknowledgements

* would like to thank Dr. Michael Alcorn and Dr. Alan Marsden for their input in designing the generators. Also the Parallel Computer Centre at Queen’s U &-entity of Belfast for the usage of the SGI Indy cluster.

References